

REVIEW AND ANALYSIS OF DENSE LINEAR SYSTEM SOLVER PACKAGE FOR DISTRIBUTED MEMORY MACHINES

by

H. N. Narang
Dept. of Computer Science
Tuskegee University
Tuskegee, AL 36088

ABSTRACT

A dense linear system solver package recently developed at the University of Texas at Austin for distributed memory machine (e.g. Intel Paragon) has been reviewed and analyzed. The package contains about 45 software routines, some written in FORTRAN, and some in C-language, and forms the basis for parallel/distributed solutions of systems of linear equations encountered in many problems of scientific and engineering nature. The package, being studied by the Computer Applications Branch of the Analysis and Computation Division, may provide a significant computational resource for NASA scientists and engineers in parallel/distributed computing.

Since the package is new and not well tested or documented, many of its underlying concepts and implementations were unclear; our task was to review, analyze, and critique the package as a step in the process that will enable scientists and engineers to apply it to the solution of their problems.

All routines in the package were reviewed and analyzed. Underlying theory or concepts which exist in the form of published papers or technical reports, or memos, were either obtained from the author, or from the scientific literature; and general algorithms, explanations, examples, and critiques have been provided to explain the workings of these programs. Wherever the things were still unclear, communications were made with the developer (author), either by telephone or by electronic mail, to understand the workings of the routines. Whenever possible, tests were made to verify the concepts and logic employed in their implementations. A detailed report is being separately documented to explain the workings of these routines.

INTRODUCTION

The solutions of linear systems of equations is needed in many science and engineering applications which include structural mechanics, fluid dynamics, chemical reactions, heat transfer, weather prediction, and climate modeling. Many physical phenomena which model a system of ordinary or partial differential equations eventually lead to the solution of a linear system of equations when discretized.

For many years, computer codes have been developed to solve these systems of equations and made available in the form of standard software packages such as LINPACK and LAPACK. Both

these packages are based on BLAS1, BLAS2, BLAS3 (Basic Linear Algebra Subprograms) which mathematicians developed over 15 years to do matrix computations [13, 20, 32, 35, 36, 38]. Whereas, LINPACK [14, 15, 37, 38, 42] provides codes for solving these systems of equations on uniprocessor machines, LAPACK [3, 7, 8, 9, 14, 15, 23, 32, 34, 37] contains the software routines to solve these systems on shared memory multiprocessors and supercomputers. Attempts have recently been made [2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 15, 16, 18, 19, 21, 22, 23, 24, 27, 28, 31, 33, 34, 40, 44]) to develop the software packages to solve these systems on distributed memory machines such as Intel's hypercubes and a network of machines. The anticipation is that future trends of the scientific computations will be on distributed memory machines and/or on network of smaller machines, and hence, this thrust in the development of codes for these architectures.

Although one can program distributed memory machines to solve scientific problems, the state of the art has not developed to the extent that the programmer does not have to be concerned about the machine architecture and its implications for algorithm selection and implementation. To advance the state of this art, as well as to apply the existing art to various scientific and engineering applications, the Langley Research Center, as part of the High Performance Computing and Communication Program, acquired an experimental machine called Paragon. This is developed by Intel Corporation of Oregon and is designed to interconnect hundreds of processors in a rectangular mesh network. Each of the processors has its own memory (32 MB), and the system can simultaneously and cooperatively solve many applications on a subset of the total number of processors (called partition) for each application [41, 43].

The machine, along with the other software, came with a software package called MPLINPACK or Multiple Processors Linear Equations Solver Package. This software has proven to be very efficient on the Paragon. Our objective was to review and analyze, the software modules to better understand the underlying theory and implementation features that are responsible for its efficiency.

There are about 45 modules written both in FORTRAN and C-languages, which can be categorized into five classes - Data Distribution Routines, Data Communication Routines, Computation Routines, Global Combination Routines, and Auxiliary Routines. These were reviewed and analyzed. Sample tests were run to validate and further understand the underlying theory and concepts for some of the modules.

In certain modules, the concepts were so new or recent that the information was only available through private technical reports or memos from the author (Professor Robert Van de Geijn at the University of Texas at Austin). In particular, the concept of TREE, CTREE in information broadcasting [17, 25, 41], Efficient Global Combine Operations [18, 19], and the implementation of FORCE-type messages [26, 43] in certain communication routines were researched and discussions made with the developer, as well as researchers in CAB and Intel Corporation scientists to understand and analyze these routines. A detailed technical report is being published and will be made available to the staff of CAB and other interested researchers.

An interesting problem came up when acceptance tests, consisting of a large system of equations (of the order of 9000×9000) were being conducted by Mr. Bulle of CAB on a rectangular grid of nodes for this machine (Paragon). It was discovered that the performance of the system was much better on an elongated grid (2×32 grid) than on its transposed grid (32×2 grid). After careful analysis, and discussions with the author (developer), we learned that the software employed Gauss elimination (row) procedure and used different techniques of information communication in one direction of the grid (across the columns) than in the other direction (down the columns). That is, the information (pivot row indices, column and row factored panels) broadcast across the column nodes is done in pipelined (ring) fashion, whereas the information (pivot element and pivot row indices) broadcast down the column nodes uses the CTREE technique. Since the CTREE broadcast technique is slower than the pipelining technique, the fewer node rows in a grid would give better performance.

ACKNOWLEDGEMENTS

The author is thankful to Mr. Richard Bulle of CAB, ACD, NASA Langley Research Center for many useful discussions, Dr. Jules Lambiotte, Head, Computer Applications Branch, NASA Langley Research Center for the opportunity to work on the project, and Dr. R. A. Van de Geijn of the Computer Science Department, University of Texas at Austin for many discussions, clarifications, and for providing the necessary literature and technical reports.

REFERENCES

1. Chu, E.; and George, A.: QR Factorization of a Dense Matrix on a Hypercube Multiprocessor; *SIAM J. Sci. Stat. Computing*, Vol. 11, No. 5, pp. 990-1028, September 1990.
2. Choi, J.; Donzarra, J. J.; and Walker, D. W.: The Design of Scalable Libraries for Distributed Memory Concurrent Computers. *Proceedings of the Workshop on Environments and Tools for Parallel Scientific Computing*, Saint Hilaire du Touvet, France, September 7-8, 1992.
3. Anderson, E., et al: Basic Linear Algebra Communication Subprograms. *Proceedings of the Sixth Distributed Memory Computing Conference*, pp. 287-290, IEEE Press, 1991.
4. Li, G. and Coleman, T. F.: A Parallel Triangular Solver for a Distributed Memory Multiprocessor. *SIAM J. Science Stat. Computing*, Vol. 9, No. 1, pp. 485-502, 1990.
5. Anderson, E. and et al: A Look at Scalable Dense Linear Algebra Libraries. J. H. Saltz, Editor. *Proceedings of the 1992 Scalable High Performance Computing Conference*, IEEE Press, 1992.
6. Anderson et al: LAPACK for Distributed Memory Architectures: Progress Report, *Parallel Processing for Scientific Computing*, Fifth SIAM Conference, SIAM Press, 1990.

7. Anderson, E., et al: LAPACK User's Guide. SIAM Press, Philadelphia, PA, 1992.
8. Anderson, E., et al: A Portable Linear Algebra Library for High Performance Computers. In Proceedings of Supercomputing, 1990, pages 1-10, IEEE Press, (also available as LAPACK Working Note #20, 1990).
9. Donzarra, J. J. and Ostrouchou, S.: LAPACK Working Note #24: Block Factorization Algorithms on the Intel iPSC/860, Tech. Report CS-90-115, University of Tennessee at Knoxville, Computer Science Department, October 1990.
10. Geijn, R. A. van de: Massively Parallel LINPACK Benchmark on the Intel Touchstone Delta and iPSC/860 Systems. Computer Science Report TR-91-28, University of Texas at Austin, 1991.
11. Dongarra, J. J.; and Van de Geijn, R. A.: Reduction to Condensed Form of Eigenvalue Problem on Distributed Memory Architectures. LAPACK Working Note #30, Tech. Report, CS-91-130, University of Tennessee at Knoxville, 1991.
12. Dongarra, J. J.: A Workshop on BLACS. LAPACK Working Note #34, Tech. Report CS-91-134, University of Tennessee, 1991.
13. Dongarra, J. J., et al: A Set of Level 3 Basic Linear Algebra Subprograms. ACM Transactions on Mathematical Software 16 (1), 1-17, 1990.
14. Dongarra, J. J. et al: Implementing Linear Algebra Algorithms for Dense Matrices on a Vector Pipeline Machine. SIAM Review, Vol. 26, #1, pp. 91-112, 1984.
15. Gallivan, K. A., et al: Parallel Algorithms for Dense Linear Algebra Computations. SIAM Review, 33, pp. 54-135, 1990.
16. Dammal, J. W., et al: Block LU Factorization. LAPACK Working Note #40, University of Tennessee, 1992.
17. Barnett, M., et al: Optimal Broadcasting in Mesh-Connected Architectures. Tech. Report, TR-91-38, University of Texas at Austin, December 4, 1991.
18. Van de Geijn, R. A. : On Global Combine Operations. LAPACK Working Note #29, Tech. Report CS-91-129, University of Tennessee, 1991.
19. Van de Geign, R. A.: Efficient Global Combine Operations. In Sixth Distributed Memory Computing Conference Proceedings, pp. 291-294, IEEE Press, 1991.
20. Choi, J., et al: Level 3 BLAS for Distributed Memory Concurrent Computers. Proceedings of the Workshop on Environments and Tools for Parallel Scientific Computing, St. Hilaire du Touvet, France, Septe. 7-8, 1992.

21. Anderson, E., et al: Performance of LAPACK: A Portable Library of Numerical Linear Algebra Routines. LAPACK Working Note #44, Tech. Report CS-92-156, University of Tennessee, 1992
22. Dongarra, J. J., et al: An Object Oriented Design for High Performance Linear Algebra on Distributed Memory Architectures. Tech. Report, Oak Ridge National Lab., 1993
23. Anderson, E., et al: Prospectus for an Extension to LAPACK: A Portable Linear Algebra Library for High Performance Computers. LAPACK Working Note #26, CS-90-118, University of Tennessee, 1991.
24. Dammal, J. and Kahan, W.: On Designing High Performance Numerical Libraries. LAPACK Working Note #39, University of Tennessee, 1991.
25. Ho, C. T., and Johnsson, S. L.: Distributed Routing Algorithms for Personalized Communications in Hypercubes. In Proceedings of 1986 International Conference on Parallel Processing, pp. 640-648, IEE Press, 1986.
26. Bokhari, S. H.: Complete Exchange on the iPSC 860, ICASE, Tech. Report 91-4, NASA LaRC, Hampton, VA, January 1991. Memory Multiprocessors. SIAM J. Sci. Stat. Computing, #3, pp. 558-587, 1988.
27. Heath, M. T., and Romine, C. H.: Parallel Solution of Triangular Systems on Distributed Memory Multiprocessors. SIAM J. Sci. Stat. Computing, #3, pp. 558-587, 1988.
28. Geist, G. A., and Romine, C. H.: LU Factorization on Distributed Memory Multiprocessor Architectures. SIAM J. Sci. Stat. Comput., Vol. 9, #4, July 1988.
29. Dongarra, J. J., et al: The IBM RISC System/6000 and Linear Algebra Operations. Oak Ridge National Lab. Tech. Report ORNL/TM 11768, pp. 13-24, Dec. 1991.
30. Gallivan, K.: Impact of Hierarchical Memory Systems on Linear Algebra Algorithm Design. Int. J. of Supercomputer Applications, Vol. 2, #1, pp. 12-48, 1989.
31. Choi, J., et al: ScaLAPACK: A Scalable Library for Distributed Memory Concurrent Computers. In Proceedings of the Fourth Symposium on the Massively Parallel Computation (Frontiers' 1992), McLean, VA, October 19-21, 1992. (Also available as LAPACK Working Note #55, University of Tennessee)
32. Dammal, J. W.; Higham, N. J.: Stability of Block Algorithms with Fast Level 3 BLAS. LAPACK Working Note #22, pp. 1-25, University of Tennessee, 1990
33. Choi, J., et al: PUMMA: Parallel Universal Matrix Multiplication Algorithms on Distributed Memory Concurrent Computers. Tech. Report, Oak Ridge National Lab, 1993.

34. Dongarra, J. J.: Two-Dimensional Linear Algebra Communication Subprograms. LAPACK Working Note #37, Tech. Report CS-91-138, University of Tennessee, 1991.
35. Lawson, C. L. et al: Basic Linear Algebra Subprograms for FORTRAN Usage. ACM Trans. Math Soft. 5, pp. 308-323, 1979.
36. Dongarra, J. J.: An Extended Set of FORTRAN Basic Linear Algebra Subprograms. ACM Trans. Math. Soft., Vol. 14, #1, pp. 1-17, 1988.
37. Ortega, James M.: Int. to Parallel and Vector Solutions of Linear Systems. Plenum Press, New York, 1988.
38. Coleman, T. E.; Loan, C. V.: Handbook for Matrix Computations. SIAM Press, Philadelphia, 1988.
39. Dongarra, J. J. and Van de Geijn, R. A.: Reduction to Condensed Form for the Eigen Value Problem on Distributed Memory Architectures. LAPACK Working Note #30, University of Tennessee, 1991.
40. Dongarra, J. J.: Performance of Various Compouters Using Standard Linear Equations Software. Tech. Report CS-89-85, University of Tennessee, 1993.
41. Paragon OSF/1, Users Guide, Intel Corporation, 1993.
42. Dongarra, J. J.: LINPACK Working Note #9, Preliminary LINPACK User's Guide, Argonne National Lab., 1977.
43. Paragon OSF/1 C System Calls, Reference Manual, Intel Corporation, 1993.
44. Cosnard, M. , et al: Parallel Gaussian Elimination on a MIMD Computer. Parallel Computing, #6, pp. 275-296, 1988.